

METHODS, APPARATUS AND DATA STRUCTURES FOR IMPOSING A
POLICY OR POLICIES ON THE SELECTION OF A LINE BY A NUMBER
OF TERMINALS IN A NETWORK

5 § 1. BACKGROUND OF THE INVENTION

§ 1.1 FIELD OF THE INVENTION

The present invention concerns accessing a packet
10 network, such as the Internet for example, over a number of
lines, such as digital subscriber lines (or "DSL lines")
for example. More specifically, the present invention
concerns the selection of one of the lines over which
access to the packet network is facilitated.

15

§ 1.2 RELATED ART

The description of art in this section is not,
and should not be interpreted to be, an admission that such
20 art is prior art to the present invention. First, since
the present invention often refers to communications
protocols, the concept of a communications protocol stack
is introduced in § 1.2.1 below for the reader's
convenience. Then, network architecture reference models
25 are introduced in § 1.2.2 below for the reader's
convenience. These two (2) sections introduce certain
concepts and terms that are well known to those skilled in
the art. Then, accessing a packet (e.g., IP) network
(e.g., the Internet) is described in § 1.2.3 below for the
30 reader's convenience. Finally, challenges faced by
customers, having at least one local area network (or LAN),
accessing the network via multiple lines, such as digital

subscriber lines (or DSL lines) for example, are described in § 1.2.4 below.

§ 1.2.1 COMMUNICATIONS PROTOCOL STACK

5

Although networking software and network reference models are known to those skilled in the art, they are introduced here for the reader's convenience.

- 10 To reduce their complexity, networks may be organized as a series of layers or levels, each one built upon the one below it as shown in Figure 1. Each layer functions to offer certain services to the higher layer, thereby shielding those higher layers from the details of
15 how the offered services are actually implemented. The entities comprising the corresponding layers on different machines are called "peers". Such peers use rules and conventions, also referred to as the layer n protocol, to communicate with each other as depicted by the dashed lines
20 in Figure 1. Actually, no data are directly transferred from layer n on one machine to layer n on another machine. Rather, each layer passes data and control information to the layer immediately below it, until the lowest layer (layer 1) is reached. Below layer 1, is a physical medium
25 110 through which actual communications take place. Thus, referring to Figure 1, actual communications take place via the solid lines and the physical medium 110, while virtual peer-to-peer communications occur via the dashed lines.

30 Still referring to Figure 1, interfaces are arranged between adjacent layers. Each of these interfaces

defines primitive operations and services that the lower layer offers to the upper layer.

The set of layers and protocols may be referred 5 to, collectively, as a "network architecture". A list of protocols used by a system, one protocol per layer, may be referred to as a "protocol stack" or "protocol suite".

S 1.2.2 NETWORK ARCHITECTURE REFERENCE MODELS

10

Figure 2 illustrates a comparison of the Open Systems Interconnection (or "OSI") reference model 210 for network architectures and the transfer control protocol/Internet protocol (or "TCP/IP") reference model 15 220 for network architectures. Although those skilled in the art will be familiar with both reference models, each is introduced below for the reader's convenience.

S 1.2.2.1 THE OSI REFERENCE MODEL

20

As shown in Figure 2, the OSI reference model 210 has seven (7) distinct layers; namely, (i) a physical layer 211, (ii) a data link layer 212, (iii) a network layer 213, (iv) a transport layer 214, (v) a session layer 215, (vi) a 25 presentation layer 216, and (vii) an application layer 217. Each layer is briefly introduced below.

The physical layer 211 deals with transmitting raw bits over a communications channel. Thus, the physical 30 layer is typically concerned with mechanical, electrical, optical, and procedural interfaces, as well as the physical transmission medium (e.g., twisted copper pair, co-axial

cable, optical fiber, etc.) which lies below the physical layer.

The data link layer 212 functions to transform a
5 raw communications facility into a line that appears free
from undetected transmission errors to the network layer
213. The data link layer 212 does this by having the
sending host segment its data into "data frames",
transmitting these frames to the receiving host, and
10 processing "acknowledgement frames" sent back from the
receiver.

The network layer 213 functions to control the
operation of a subnetwork between the hosts and controls
15 the routing of packets between the hosts.

The transport layer 214 functions to accept data
from the session layer 215 and segment this data into
smaller units, if necessary, for use by the network layer
20 213. The transport layer 214 also determines a type of
service (e.g., an error-free, point-to-point service) to
provide to the session layer 215. Further, the transport
layer 214 controls the flow of data between hosts. The
transport layer 214 is a true "end-to-end" layer, from
25 source host to destination host, since a program on the
source machine converses with a similar program on the
destination machine, using message headers and control
messages.

30 The session layer 215 functions to allow
different machines to establish sessions between them. The

session layer 215 may manage dialog control and maintain synchronization.

The presentation layer 215 concerns the syntax
5 and semantics of information transmitted.

The application layer 216 may function to define network virtual terminals that editors and other programs can use, and to transfer files.

10

S 1.2.2.2 THE TCP/IP MODEL

In recent decades, and in the past five (5) to ten (10) years in particular, computers have become
15 interconnected by networks by an ever increasing extent; initially, via local area networks (or "LANs"), and more recently via LANs, wide area networks (or WANs) and the Internet. In 1969, the Advanced Research Projects Agency (ARPA) of the U.S. Department of Defense (DoD) deployed
20 ARPANET as a way to explore packet-switching technology and protocols that could be used for cooperative, distributed, computing. Early on, ARPANET was used by the TELNET application which permitted a single terminal to work with different types of computers, and by the file transfer
25 protocol (or "FTP") which permitted different types of computers to transfer files from one another. In the early 1970s', electronic mail became the most popular application which used ARPANET.

30 Since this packet switching technology was so successful, the ARPA applied it to tactical radio communications (Packet Radio) and to satellite

communications (SATNET). However, since these networks operated in very different communications environments, certain parameters, such as maximum packet size for example, were different in each case. Thus, methods and 5 protocols were developed for "internetworking" these different packet switched networks. This work lead to the transmission control protocol (or "TCP") and the internet protocol (or "IP") which became the TCP/IP protocol suite. Although the TCP/IP protocol suite, which is the foundation 10 of the Internet, is known to those skilled in the art, it is briefly described below for the reader's convenience.

As shown in Figure 2, the TCP/IP reference model 220 includes a physical layer 221, a network access layer 15 222, an internet layer 223, a transport layer 224, and an application layer 225. Each of these layers is briefly introduced below.

The physical layer 221 defines the interface 20 between a data transmission device (e.g., a computer) and a transmission medium (e.g., twisted pair copper wires, co-axial cable, optical fiber, etc.). It specifies the characteristics of the transmission medium, the nature of the signals, the data rate, etc.

25

The network access layer 222 defines the interface between an end system and the network to which it 30 is attached. It concerns access to, and routing data across, a network. Frame relay is an example of a network access layer.

The internet layer 223 functions to permit hosts to inject packets into any network and have them travel independently to the destination machine (which may be on a different network). Since these packets may travel 5 independently, they may even arrive in an order other than the order in which they were sent. Higher layers can be used to reorder the packets. Thus, the main function of the internet layer 320 is to deliver (e.g., route) IP packets to their destination.

10

The transport layer 224 is an end-to-end protocol. For example, the transmission control protocol (or "TCP") is a reliable connection-oriented protocol that allows a byte stream originating on one machine to be 15 delivered, without error, on any other machine on the Internet. More specifically, the TCP protocol fragments an incoming data stream into discrete messages, each of which is passed to the internet layer 223. At the destination, the TCP protocol reassembles the received messages into an 20 output stream.

The TCP/IP model 220 does not have session and presentation layers. Instead, an application layer 225 contains all of the higher-level protocols which are used 25 to support various types of end use applications (e.g., the simple mail transfer protocol (or "SMTP") for e-mail, the file transfer protocol (or "FTP"), etc.).

The TCP/IP model does not define what occurs 30 below the internet layer 223, other than to note that the host has to connect to the network using some protocol so

that it can send IP packets over it. This protocol varies from host to host and network to network.

Basically, each of the layers encapsulates, or
5 converts, data in a higher level layer. For example,
referring to Figure 4, user data 400 as a byte stream is
provided with a TCP header 402 to form a TCP segment 410.
The TCP segment 410 is provided with an IP header 412 to
form an IP datagram 420. The IP datagram 420 is provided
10 with a network header 422 to define a network-level packet
430. The network-level packet 430 is then converted to
radio, electrical, optical (or other) signals sent over the
transmission medium at a specified rate with a specified
type of modulation.

15 The TCP header 402, as illustrated in Figure 5,
includes at least twenty (20) octets (i.e., 160 bits).
Fields 502 and 504 identify ports at the source and
destination systems, respectively, that are using the
connection. Values in the sequence number 506,
acknowledgement number 508 and window 516 fields are used
to provide flow and error control. The value in the
checksum field 518 is used to detect errors in the TCP
segment 410.

25 Figures 6A and 6B illustrate two (2) alternative IP headers 412 and 412', respectively. Basically, Figure 6A depicts the IP protocol (Version 4) which has been used. Figure 6B depicts a next generation IP protocol (Version 6) which, among other things, provides for more source and destination addresses.

30

More specifically, referring to Figure 6A, the four (4) bit version field 602 indicates the version number of the IP, in this case, version 4. The four (4) bit Internet header length field 604 identifies the length of 5 the header 412 in 32-bit words. The eight (8) bit type of service field 606 indicates the service level that the IP datagram 420 should be given. The sixteen (16) bit total length field 608 identifies the total length of the IP datagram 420 in octets. The sixteen (16) bit 10 identification field 610 is used to help reassemble fragmented user data carried in multiple packets. The three (3) bit flags field 612 is used to control fragmentation. The thirteen (13) bit fragment offset field 614 is used to reassemble a datagram 420 that has become 15 fragmented. The eight (8) bit time to live field 616 defines a maximum time that the datagram is allowed to exist within the network it travels over. The eight (8) bit protocol field 618 defines the higher-level protocol to which the data portion of the datagram 420 belongs. The 20 sixteen (16) bit header checksum field 620 permits the integrity of the IP header 412 to be checked. The 32-bit source address field 322 contains the IP address of the sender of the IP datagram 420 and the 32-bit destination address field contains the IP address of the host to which 25 the IP datagram 120 is being sent. Options and padding 626 may be used to describe special packet processing and/or to ensure that the header 412 is a complete multiple of 32-bit words.

30 Referring to Figure 6B, the four (4) bit version field 602 indicates the version number of the IP, in this case, version 6. The four (4) bit priority field 628

enables a sender to prioritize packets sent by it. The 24-bit flow label field 630 is used by a source to label packets for which special handling is requested. The sixteen (16) bit payload length field 632 identifies the 5 size of data carried in the packet. The eight (8) bit next header field 634 is used to indicate whether another header is present and if so, to identify it. The eight (8) bit hop limit field 636 serves to discard the IP datagram 420 if a hop limit (e.g., the number of times the packet is 10 routed) is exceeded. Also provided are 128-bit source and destination address fields 322' and 324', respectively.

Having described the TCP/IP protocol stack 220, the routing of a TCP/IP packet is now described.

15 A TCP/IP packet is communicated over the Internet (or any internet or intranet) via routers. Basically, routers in the Internet use destination address information (Recall fields 624 and 624'.) to forward packets towards 20 their destination. Routers interconnect different networks. More specifically, routers accept incoming packets from various connected networks, use a look-up table to determine a network upon which the packet should be placed, and routes the packet to the determined network.

25 Figure 7, which includes Figures 7A through 7C, illustrates the communication of data from a sender, to a receiver, using the TCP/IP protocol stack. Referring first to Figure 7A, an application protocol 702 prepares a block 30 of data (e.g., an e-mail message (SMTP), a file (FTP), user input (TELNET), etc.) 400 for transmission. Before the data 400 are sent, the sending and receiving applications

agree on a format and encoding and agree to exchange data (Recall, e.g., the peer-to-peer communications depicted with dashed lines in Figure 1.). If necessary, the data are converted (character code, compression, encryption, etc.) to a form expected by the destination device.

The TCP layer 704 may segment the data block 400, keeping track of the sequence of segments. Each TCP segment 410 includes a header 402 containing a sequence number (recall field 506) and a frame check sequence to detect errors. A copy of each TCP segment is made so that if a segment is lost or damaged, it can be retransmitted. When an acknowledgement of safe receipt is received from the receiver, the copy of the segment is erased.

The IP layer 706 may break the TCP segment into a number of datagrams 420 to meet size requirements of networks over which the data will be communicated. Each datagram includes the IP header 412.

A network layer 708, such as frame relay for example, may apply a header and trailer 422 to frame the datagram 420. The header may include a connection identifier and the trailer may contain a frame check sequence for example. Each frame 430 is then transmitted, by the physical layer 710, over the transmission medium as a sequence of bits.

Figure 7B illustrates the operation of the TCP/IP protocol stack at a router in the network. The physical layer 712 receives the incoming signal 430 from the transmission medium and interprets it as a frame of bits.

The network (e.g., frame relay) layer 714 then removes the header and trailer 422 and processes them. A frame check sequence may be used for error detection. A connection number may be used to identify the source. The network
5 layer 714 then passes the IP datagram 420 to the IP layer 718.

The IP layer examines the IP header 412 and makes a routing decision (Recall the destination address 324,
10 324'). A logical link control (or "LLC") layer 720 uses a simple network management protocol (or "SNMP") and adds a header 750 which contains a sequence number and address information. Another network layer 722 (e.g., media access control (or "MAC")) adds a header and trailer 760. The
15 header may contain address information and the trailer may contain a frame check sequence. The physical layer 724 then transmits the frame 450 over another transmission medium.

20 Figure 7C illustrates the operation of the TCP/IP protocol stack at a receiver. The physical layer 732 receives the signals from the transmission medium and interprets them as a frame of bits. The network layer 734 removes the header and trailer 760 and processes them. For
25 example, the frame check sequence in the trailer may be used for error detection. The resulting packet 440 is passed to the transport layer 736 which processes the header 750 for flow and error control. The resulting IP datagram 420 is passed to the IP layer 738 which removes
30 the header 412. Frame check sequence and other control information may be processed at this point.

The TCP segment 410 is then passed to the TCP layer 740 which removes the header 402 and may check the frame check sequence. (In the event of a match, the match is acknowledged and in the event of a mismatch, the packet 5 is discarded.) The TCP layer 740 then passes the data 400 to the application layer 742. If the user data was segmented (or fragmented), the TCP layer 740 reassembles it. Finally, the application layer 742 performs any necessary transformations, such as decompression and 10 decryption for example, and directs the data to an appropriate area of the receiver, for use by the receiving application.

DOCUMENT NUMBER

Having described the concept of a network
15 architecture, as well as the OSI reference model and the TCP/IP protocol suite, certain background technologies, all related to accessing a packet network, are now introduced in § 1.2.3 below.

20 **S 1.2.3 PACKET (e.g., INTERNET OR IP) NETWORK
 ACCESS**

The present invention is related to accessing a packet network such as the Internet for example. To
25 appreciate at least some of the advantages of the present invention, certain technologies associated with accessing a packet network should be understood. Although these technologies are understood by those skilled in the art, they are introduced below for the reader's convenience.
30 More specifically, the point-to-point protocol (or "PPP") is introduced in § 1.2.3.1 below, local area networks and

Ethernet are introduced in § 1.2.3.2 below, digital subscriber line (or "DSL") services are introduced in § 1.2.3.3 below, and the point-to-point protocol over Ethernet (PPPoE) protocol is introduced in § 1.2.3.4 below.

5

§ 1.2.3.1 POINT-TO-POINT PROTOCOL ("PPP")

Briefly stated, the point-to-point protocol (or ("PPP")) is a data link layer (i.e., layer 2) protocol that
10 allows a computer to connect to the Internet via a standard dial-up telephone line and a modem. Typically, the computer will call up an Internet service provider (or "ISP"). The computer may merely function as a character-oriented terminal logged into the Internet
15 service provider's time-sharing system. In this mode, also referred to as a "shell account", a user can type commands and run programs, but graphical Internet services, such as the World Wide Web for example, are not available. On the other hand, a computer can call the Internet service
20 provider's router and act like a full-blown Internet host, in which case, all Internet services, including graphical services, are available.

The serial line IP (or "SLIP") protocol was the
25 first widely used point-to-point data link protocol. In accordance with the SLIP protocol, a computer would just send raw IP packets over the line, with a special framing flag byte at the end. Unfortunately, SLIP has a number of serious problems. First, it doesn't detect or correct
30 errors -- higher layers are left to deal with these functions. Second, SLIP supports only IP. Third, each party's IP address must be known by the other in advance.

That is, neither IP address can be dynamically assigned during setup. This third limitation precludes, as a practical matter, the use of SLIP by various Internet service providers since they often dynamically assign a number of Internet addresses to their customers (so that they can service a number of customers which is greater than the number of Internet addresses that they have). Fourth, SLIP does not support authentication. Finally, since SLIP is not an approved Internet standard, many different and incompatible versions exist.

The point-to-point protocol (or "PPP") was designed to overcome the problems of the SLIP protocol. That is, among other improvements, PPP detects errors, supports multiple protocols, permits IP addresses to be negotiated at connection time, and permits authentication. Basically, PPP provides (1) a framing method that unambiguously delineates the start and end of frames and that handles error detection, (2) a link control protocol (or "LCP") for bringing lines up, testing the times, negotiating options, and bringing the lines down, and (3) a way to negotiate network-layer options independently of the network layer protocol to be used.

Figure 8 illustrates a PPP full frame format 800. This frame 800 may use character stuffing to ensure that all frames are an integral number of bytes. The frame 800 starts with a standard high-level data link control (or "HDLC") flag byte "01111110", denoted 810. An address field 820 is typically always set to "11111111" to indicate that all stations are to accept the frame. A control field 830 may be used to number a sequence of frames, although a

default value of "00000011" indicates an unnumbered frame. Note that since, in a default mode, the address and control fields 820 and 830 are constant, two parties may negotiate to omit these fields, thereby saving two (2) bytes per
5 frame, via the link control protocol (or "LCP"). A protocol field 840 identifies what type of packet (e.g., LCP, NCP, IP, IPX, AppleTalk, etc.) is in the payload field 850. Codes starting with a "0" bit are network layer protocols (e.g., IP, IXP, OSI, CLNP, XNS), while codes
10 starting with a "1" bit are protocols (e.g., LCP, NCP) used to negotiate other protocols. The default size of the protocol field 840 is two (2) bytes. However, the parties
can negotiate this field down to one (1) byte per frame, again via the link control protocol (or "LCP"). A payload
15 field 850 has a variable length, up to a negotiated maximum value. If the length is not negotiated during line setup (e.g., via LCP), a default length of 1500 bytes is used. A checksum field 860 is used for error detection. The checksum field 860 has a default value of two (2) bytes,
20 but can be negotiated to four (4) bytes. A standard high-level data link control (HDLC) flag byte "01111110", denoted 870, ends the frame 800.

Now, an exemplary operation of the PPP, in a
25 situation where a user calls up an Internet service provider to make their computer a temporary Internet host, is described. The computer calls the Internet service provider's router via a modem. The router's modem then answers the call and establishes a physical connection.
30 The computer then sends the router a series of LCP packets in the payload field 850 of one or more PPP frames 800. These packets and the responses from the router negotiate

the PPP parameters to be used during the course of the call. Then, the computer sends a series of NCP packets to configure the network layer. For example, a computer typically wants to run the TCP/IP protocol stack and

5 therefore needs an IP address. Since there are not enough IP addresses for each user to have a static, permanent, IP address, each Internet service provider typically obtains a number of IP addresses and dynamically assigns them to each attached device for the duration of a session with the

10 attached device. In this way, the Internet service provider can have more customers than IP addresses (but cannot service all of its customers simultaneously). The NCP for IP is used to assign the IP address to the computer. At this point, the computer is an Internet host

15 and can therefore send and receive IP packets, just as hardwired hosts can. When the use is done, the NCP is used to tear down the network layer connection, thereby freeing up the assigned IP address. The LCP is then used to terminate the data link layer connection. Finally, the

20 computer instructs its modem to hang up the phone line, thereby releasing the physical layer connection. Figure 9 illustrates a sequence of states that the PPP may go through when bringing up and tearing down (modem or router-router) line connections.

25 *Sub C1* Having described how the point-to-point protocol may be used to establish a connection from a computer to the Internet, it should be noted that many computers are connected to a local area network (or LAN) and access the

30 Internet via the LAN. Section 1.2.3.2 below introduces LANs in general, and Ethernet in particular.

S 1.2.3.2 LOCAL AREA NETWORKS (LANs) AND ETHERNET

Although those skilled in the art understand LANs
5 and Ethernet, each is introduced here for the reader's convenience.

Local area networks (or "LANs") have been used to connect computers in offices, schools and factories to
10 share resources (e.g., printers) and to exchange information. LANs are generally restricted in size. Thus, the worst-case transmission time is bounded and known in advance. Knowing this bound permits certain network designs, that would not be possible otherwise, to be used,
15 and also simplifies network management. LANs typically use a single cable to which all machines are attached for transmitting communications amongst the machines. LANs may employ various topologies, such as bus, ring, etc.

20 Ethernet is a well known and widely deployed local area network (or "LAN") protocol. Ethernet has a bus (as opposed to a ring or star) topology. Devices on an Ethernet LAN can transmit whenever they want to -- if two (2) or more packets collide, each device waits a random
25 time and tries again. More specifically, as defined in IEEE 802.3, Ethernet is a LAN with persistent carrier sense multiple access (or "CSMA") and collision detection (or "CD"). If a device wants to transmit, it "listens" to the cable (hence the term "carrier sense"). If the cable is
30 sensed as being busy, the device waits for the cable to become idle. If the cable is idle or becomes idle, the device transmits. If two (2) or more devices begin to

transmit simultaneously (hence the term "multiple access"), there will be a collision which will be detected (hence the term "collision detection"). In the event of a collision, the devices causing the collision will (i) terminate their
5 transmission, (ii) wait a random time, and (iii) try to transmit again (assuming that the cable is idle). Accordingly, a CSMA/CD cable or bus has one (1) of three (3) possible states -- contention (or collision), transmission, or idle.

10

The IEEE 802.3 frame structure 1000 (or MAC Sublayer Protocol) is illustrated in Figure 12. Each frame 1200 starts with a preamble 1210 of seven (7) bytes (each byte containing the pattern "10101010"). The Manchester 15 encoding (which defines a "1" with a high-to-low signal transition and a "0" with a low-to-high signal transition) of this pattern produces a 10-MHz square wave for 5.6 μ sec to allow the receiver's clock to synchronize with the sender's. The one (1) byte start of frame delimiter 1220 20 contains the pattern "10101011" to denote the start of the frame. The source and destination addresses 1230 and 1240, respectively, may be six (6) bytes (or 48 bits) long. The second most significant bit is used to distinguish local addresses from global addresses. Thus 46 bits are 25 available for addresses (or about 7×10^{13} unique addresses. Thus, any device can uniquely address any other device by using the right 48-bit address -- it is up to the network layer to figure out how to locate the device associated with the destination address.

30

These 48-bit source and destination addresses

1230 and 1240, respectively, may be referred to as media access control (or "MAC") addresses. Basically, each device that may be connected to a network or the Internet has an assigned unique MAC address. (Some bits of the MAC 5 address are assigned to various device manufactures. The manufactures then ensure that each device manufactured by it has a unique MAC address.)

The two (2) byte length of data field 1250
10 indicates the number of bytes (between 0 and 1500) present in the data field 1260. Valid frames 1200 must be at least 64 bytes long. Thus, if the data field 1260 is less than 46 bytes, the pad field 1270 is used to ensure that the frame 1200, from the destination address field 1230 through 15 the checksum field 1280, is at least 64 bytes.

The four (4) byte checksum field 1280 is basically a 32-bit hash code of the data and can be used to detect errors in the data.

Having described how the point-to-point protocol
20 may be used to establish a connection from a computer to the Internet in § 1.2.3.1 above, as well as LANs in general, and Ethernet in particular in § 1.2.3.2 above, a physical layer protocol, using standard twisted pair copper 25 telephone lines, is introduced in § 1.2.3.3 below.

S 1.2.3.3 DIGITAL SUBSCRIBER LINE ("DSL")
SERVICE

Although those skilled in the art understand
5 digital subscriber line (or "DSL") services, they are
introduced here for the reader's convenience.

Voice grade data modems are presently limited to
approximately 56 Kbps. Bandwidth limitations of voice band
10 lines often do not come from the subscriber line itself.
Rather, they come from filters at the edge of the core
network which limit voice grade bandwidth to about 3.3 or 4
kHz. Without such filters, copper access lines can pass
frequencies into MHz regions, albeit with substantial
15 attenuation. Such attenuation increases with line length,
modulation frequency, and decreasing wire diameter (or
increasing wire gauge).

Digital subscriber line (or "DSL") is a generic
20 name for a group of digital services to be provided by
local telephone companies to their local subscribers. DSL
lines can carry both voice and data signals at the same
time, in both directions, as well as signaling and call
information data. High data rate DSL (or "HDSL") uses
25 advanced modulation techniques to transmit 1.544 Mbps in
bandwidths ranging from 80 kHz to 240 kHz. Such rates may
be supported over 24 gauge lines up to 12,000 feet. Single
line DSL (or "SDSL") is basically a single line version of
HDSL. (Note that "SDSL" has also been used to refer to
30 symmetric DSL.) Asymmetric DSL (or "ADSL") supports an
asymmetric data stream, as its name implies, with much more
bandwidth made available to a customer than from a

customer. For example, downstream (i.e., to customer) rates of 1.544 Mbps, 2.048 Mbps, 6.312 Mbps and 8.448 Mbps may be supported on lines up to 18,000 feet, 16,000 feet, 12,000 feet, and 9,000 feet, respectively. Upstream (i.e., from customer) rates from 16 Kbps to 640 Kbps may also be supported. Both upstream and downstream data communications operate at frequencies above that of the plain old telephone service (or POTS), such that POTS service is independent of ADSL data services.

10

As can be appreciated from this sampling of digital service line (or "DSL") services, they are expected to be very popular, particularly for Internet access.

Figure 10 illustrates two (2) customers 1010 which use a DSL service to connect with an Internet service provider 1050. The first customer has a PC and a DSL modem 1020a. The DSL modem 1020a communicates with a digital subscriber line multiplexer (or "DSLAM") 1030 via a DSL line. The DSLAM 1030 forwards communications to the Internet service provider 1050 via a network 1040. The second customer 1010b has a number of computers in a local area network (LAN) defined by the Ethernet hub 1015. The hub 1015 is coupled with a DSL modem 1020a.

25

§ 1.2.3.4 POINT-TO-POINT PROTOCOL OVER ETHERNET

As described in § 1.2.3.2 above, in many access technologies, the most cost effective method to attach multiple hosts to the customer premise access device, is via Ethernet. PPP over Ethernet (or "PPPoE") enables a network of hosts to connect, over a simple bridging access device,

to a remote access concentrator. With PPPoE, each host uses it's own PPP stack and the user is presented with a familiar user interface.

5 PPPoE has two (2) stages -- a discovery stage and a PPP session stage. When a host wishes to initiate a PPPoE session, it first performs a discovery to identify the Ethernet MAC address of the peer and to establish a PPPoE SESSION_ID. In the discovery stage, a host (the client)
10 discovers an access concentrator (the server). Based on the network topology, there may be more than one access concentrator that the host can communicate with. The discovery stage allows the host to discover all access concentrators and then select one. When the discovery stage
15 completes successfully, both the host and the selected access concentrator have the information they will use to build their point-to-point connection over Ethernet. The discovery stage remains stateless until a PPP session is established. Once a PPP session is established, both the
20 host and the access concentrator allocate the resources for a PPP virtual interface.

Since a PPPoE frame is very similar to the standard Ethernet frame 1200, the following will refer to
25 the Ethernet frame 1200 of Figure 12. Referring once again to the Ethernet frame 1200 of Figure 12, the destination address field 1230 contains either a unicast Ethernet destination address, or the Ethernet broadcast address (0xffffffff). For discovery stage packets, the value is
30 either a unicast or broadcast address. For the PPP session stage packets, this field 1230 contains the peer's unicast address (as determined from the discovery stage). The

source address field 1240 contains the Ethernet MAC address of the source device. The length of data field 1250 is used as an ether type field 1250' in the context of PPPoE (this is the main difference between Ethernet and PPPoE frames)

- 5 and may be set to either 0x8863 to indicate a discovery stage frame, or 0x8864 to indicate a PPP session stage frame.

The Ethernet payload 1260' for PPPoE is

- 10 illustrated in Figure 13. The version field 1310 is four (4) bits and may be set to 0x1, for example for a particular version of the PPPoE specification. The type field 1320 is four (4) bits and may be set to 0x1, for example for a particular version of the PPPoE specification. The code field is eight (8) bits and may be used to identify certain type of information used in the discovery and PPP session stages. The session ID field 1350 is sixteen (16) bits and is an unsigned value in network byte order. Its value is defined for discovery stage packets and is fixed for a given 15 PPP session. The value of the session ID field 1350, as well as the values for the source and destination address fields 1230 and 1240, may be used to uniquely identify a PPPoE session. The length field 1360 is sixteen (16) bits. Its value, in network byte order, indicates the length of 20 the PPPoE payload 1370.

Having introduced the PPPoE protocol and the PPP payload for PPPoE, the stages of the PPPoE protocol are now described. The discovery stage includes four (4) steps:

- 30 (i) the Host broadcasting an initiation packet; (ii) one or more access concentrators sending offer packets; (iii) the host sending a unicast session request packet; and (iv) the

selected access concentrator sending a confirmation packet. Once the host receives the confirmation packet, each of the peers know the PPPoE session ID and each other's Ethernet address. The host may proceed to the PPP session stage.

- 5 When the access concentrator sends the confirmation packet, it may proceed to the PPP session stage.

All discovery stage Ethernet frames have the ether type field 1250' set to the value 0x8863. The PPPoE payload 10 1370 contains zero or more TAGs. A TAG is a TLV (type-length-value) construct and is illustrated in Figure 14. The TAG type field 1410 is sixteen (16) bits in network byte order. The TAG length field 1420 is sixteen (16) bits. It is an unsigned number in network byte order, indicating 15 the length in octets of the TAG value field 1430.

The discovery stage Ethernet packets, corresponding to the four (4) steps introduced above, are now described. Recall that in the first step, the host 20 broadcasts an initialization packet. The initialization packet is referred to as a PPPoE Active Discovery Initiation (or "PADI") packet. The Host sends the PADI packet with the destination address 1230 set to the broadcast address. The code field 1340 is set to 0x09 and the session ID field 1350 25 is set to 0x0000. The PADI packet contains one TAG of having a Service-Name tag type 1410, indicating the service the host is requesting, and any number of other TAG types. Figure 11 illustrates an exemplary PADI packet.

30 Recall that in the second step of the discovery stage, one or more access controllers send offer packets. An offer packet is referred to as a PPPoE Active Discovery

Offer (or "PADO") packet. More specifically, when an access concentrator receives a PADI packet that it can serve, it replies by sending a PADO packet. The destination address 1230 is simply the unicast address of the host that sent the 5 PADI packet. The code field 1340 is set to 0x07 and the session ID field 1350 is set to 0x0000. The PADO packet contains one AC-Name TAG containing the access concentrator's name, a Service-Name TAG identical to the one in the PADI packet, and any number of other Service-Name 10 TAGs indicating other services that the access concentrator offers. If the access concentrator cannot serve the PADI packet it does not respond with a PADO packet.

Recall that in the third step of the discovery 15 stage, the host sends a unicast session request packet. This packet may be referred to as a PPPoE Active Discovery Request (or "PADR") packet. The host that sent the PADI packet may receive more than one PADO packets in response. The host chooses one of these PADO packets. (The choice can 20 be based on the AC-Name or the services offered by the access concentrator.) The host then sends one PADR packet to the access concentrator selected. The destination address field 1230 is set to the unicast Ethernet address of the selected access concentrator. The code field 1340 is 25 set to 0x19 and the session ID field 1350 is set to 0x0000. The PADR packet contains one TAG of TAG_TYPE Service-Name, indicating the service the host is requesting, and any number of other TAG types.

30 Finally, recall that in the fourth step of the discovery stage, the selected access concentrator sends a confirmation packet. This confirmation packet may be

referred to as a PPPoE Active Discovery Session-confirmation (or "PADS") packet. More specifically, when the selected access concentrator receives a PADR packet, it prepares to begin a PPPoE session. It generates a unique session ID 5 value for the PPPoE session and replies to the host with a PADS packet. The destination address field 1230 is the unicast Ethernet address of the host that sent the PADR. The code field 1340 is set to 0x65 and the session ID field 1350 is set to the unique value generated for this PPPoE 10 session. The PADS packet contains one TAG of TAG_TYPE Service-Name, indicating the service under which access concentrator has accepted the PPPoE session, and any number of other TAG types. If the access concentrator does not like the Service-Name in the PADR packet, will then reply 15 with a PADS packet containing a TAG of TAG_TYPE Service-Name-Error (and any number of other TAG types). In this case the session ID field 1350 is set to 0x0000.

A PPPoE Active Discovery Terminate (or "PADT") 20 packet may be sent anytime after a session is established to indicate that a PPPoE session has been terminated. It may be sent by either the host or the access concentrator. The destination address field 1230 is a unicast Ethernet address, the code field 1340 is set to 0xa7, and the session 25 ID field 1350 be set to identify the session which is to be terminated. No TAGs are required. When a PADT is received, no further PPP traffic is allowed to be sent using that session.

As stated above, once the discovery stage is 30 complete, the PPP session stage commences. Once the PPPoE session begins, PPP data is sent as in any other PPP

encapsulation. All Ethernet packets are unicast. The Ether type field 1250' is set to 0x8864. The PPPoE code field 1340 is set to 0x00. The session ID field 1350 is the value assigned (by the access concentrator) in the fourth step 5 discovery stage. The PPPoE payload contains a PPP frame. The frame begins with the PPP Protocol-ID.

§ 1.2.4 CHALLENGES FACED BY CUSTOMERS HAVING
LAN(s) ACCESSING MULTIPLE DSL LINES

10 Currently, modem development to support digital subscriber line (or "DSL") services assumes that a single DSL line is extended from a communications provider (such as a regional bell operating company or "RBOC") to a 15 customer's premises. (Recall, for example, customers 1010 in Figure 10.) In some instances, often when the customer premises supports a small business or home office, a single DSL line may not provide sufficient bandwidth for the customer's needs. Instead of buying a dedicated line 20 (i.e., T1 or T3 connection), some customers may find that purchasing multiple DSL lines to their premises is more affordable. Many of such customers will have local area networks (or LANs).

25 Figure 15 is a high level block diagram which illustrates an environment 1500 in which a customer, having a local area network, has more than one digital subscriber line 1560 to a facility 1520 of a communications provider. The customer will often want to allow one of their 30 terminals 1512, such as a computer for example, to call the Internet service provider's router and act like a full-blown Internet host. To reiterate, such a connection

may be established in accordance with the point-to-point protocol, as described in § 1.2.3.1 above, or more specifically, via the point-to-point protocol over Ethernet protocol (or PPPoE) described in § 1.2.3.4 above.

5

As Figure 15 shows, the premises 1510 of the customer may have a LAN, such as an Ethernet LAN, defined by a Ethernet hub or bridge 1514 having a number of terminals 1512, such as computers for example, connected to it. The Ethernet hub or bridge 1514 may also be coupled with a number of ADSL terminating units-remote (or "ATU-Rs") 1516. If an Ethernet hub 1514 is used, the hub 1514 merely provides a simple means of connecting the terminals 1512 of the LAN. If an Ethernet bridge is used, it may learn the layer 2 (e.g., MAC) addresses of the terminals 1512 connected to it.

ADSL terminating units-remote or ("ATU-R") 1516 support the ADSL service and has a matching unit, an ADSL terminating unit-central office (or "ATU-C") 1522 located at a facility, such as a central office, of a communications provider 1520. An ATU-R 1516 and ATU-C 1522 pair, in combination, support a high data rate over standard copper telephone wires 1160.

25

At the communications provider facility 1520, the DSLAM containing the ATU-C units 1522 may be connected with a bridge 1524. The bridge 1524 may be used to connect the communications provider facility 1520 with an Internet service provider server 1540. The Internet service provider server 1540 should support point-to-point protocol over Ethernet (or "PPPoE") sessions. Note that the

terminals 1512 should also support PPPoE session (using software often referred to as "SHIMs").

Unfortunately, using the point-to-point protocol over Ethernet (or "PPPoE") protocol does not allow a straightforward transition from a single DSL line to multiple DSL lines. More specifically, the PPPoE protocol assumes an Ethernet infrastructure. Referring to Figure 15, each terminal (such as a personal computer for example) 1112 in the customer premises 1510 would possess an Ethernet network interface card (or "NIC"), and all terminals 1512 would connect to an Ethernet hub or Ethernet bridge 1514. An ADSL terminating unit-remote (or "ATU-R") 1516 would also connect to the Ethernet hub or learning bridge 1514. With current implementations of PPPoE, an ATU-R, which acts as learning bridge, is used to provide connectivity to the DSL network. This creates a problem for customers that want to use multiple DSL lines. More specifically, using multiple ATU-Rs connected to the Ethernet hub or Ethernet bridge may cause traffic loops (described in § 1.2.2.1 below) and race conditions (described in § 1.2.2.2 below) that may often prevent the customer from getting the bandwidth expected. Further, presently, users cannot implement any policies regarding which lines 1560 serve which terminals 1512.

§ 1.2.2.1 LOOP CONDITIONS

Recall from § 1.2.3.4 above that during a discovery stage of PPPoE, when a host (e.g., a terminal 1512) wants to establish a connection using PPPoE, it broadcasts a PPPoE Active Discovery Initiation (or "PADI")

packet and one or more access concentrators may respond with a PPPoE Active Discovery Offer (or "PADO") packet. Due to the broadcast nature of the PPPoE Active Discovery Initiation (PADI) packets, each one of the ATU-R units 1516 5 will forward these packets and, consequently, receive PPPoE Active Discovery Offer (PADO) packets in response. When a response is received via one ATU-R, situations can result where the response is actually looped back to the other ATU-R. This would occur if the user's Ethernet hub is a 10 repeatered hub which sends all Ethernet frames it receives on one port out every other port. In turn, the Ethernet frame will cause the ATU-R which inadvertently receives the Ethernet frame to update its bridging table so that it believes the ISP's access router is now on the home LAN.

15 This follows since it examines the Ethernet frame and sees a source MAC address that belongs to the ISP's router arriving on the port it uses to connect to the repeatered hub. This means that no traffic will flow through this ATU-R until such time as it receives an Ethernet frame, via 20 the DSL line that it terminates, from the ISP's router. Hence, the ability to utilize a second DSL line can be significantly hampered by the looping of received traffic from the ISP.

25

S 1.2.2.2 RACE CONDITIONS

Sub A 27

As just stated in § 1.2.2.1 above, during a discovery stage of PPPoE, when a host (e.g., a terminal 1512) wants to establish a connection using PPPoE, it 30 broadcasts a PPPoE Active Discovery Initiation (or "PADI") packet and one or more access concentrators may respond with a PPPoE Active Discovery Offer (or "PADO") packet. As

further just stated above, due to the broadcast nature of the PPPoE Active Discovery Initiation (PADI) packets, each one of the ATU-R units 1516 may forward these packets and, consequently. It may turn out that one of the ATU-R units 1516 may always receive a PADI packet response first.

Consequently, traffic between a customer's premises

Consequently, traffic between a customer's premises 1510 and its communications provider facility 1520 may not be balanced across the lines 1560. Indeed, a single line 1560 may end up serving a large amount of traffic while the other lines 1560 sit idle. Such a situation is clearly undesirable.

§ 2. SUMMARY OF THE INVENTION

15 The present invention discloses a device that can enforce user defined and/or configurable policies to select which DSL line a PPPoE session would use when there are multiple DSL lines present. This device avoids the adverse network conditions, introduced in § 1.2.2 above, that exist
20 if the multiple DSL lines are directly connected to the premises local area network (or "LAN").

Once a DSL line is selected, the PPPoE session may be supported, in its entirety, over that selected DSL line.

In one embodiment disclosed, the device may include embedded DSL modems so external ATU-Rs would not be required. Furthermore, the complexity of the modem functionality would be reduced as the embedded modem would not require the learning bridge logic, which is currently implemented in external PPPoE capable modems. No changes

to premises networking equipment (hubs, bridges, PCs) would be required to use this device.

S 3. BRIEF DESCRIPTION OF THE DRAWINGS

5

Figure 1 illustrates the way in which network communications schemes may be described by a stack of protocols.

10 Figure 2 compares the OSI reference model and the TCP/IP protocol suite.

Figure 3 illustrates internet protocol (or "IP") global addressing.

15 Figure 4 illustrates the manner in which data is encapsulated by a TCP header, an IP header, and a network header in accordance with the TCP/IP protocol suite.

20 Figure 5 illustrates the fields of a TCP header.

Figures 6A and 6B illustrate the fields of Version 4 and Version 6, respectively, of the IP header.

25 Figures 7A through 7C illustrate the transmission of data over a network in accordance with the TCP/IP protocol suite.

30 Figure 8 illustrates a point-to-point protocol full frame format.

Figure 9 is a simplified phase diagram for bringing a connection up and down in accordance with the point-to-point protocol.

5 Figure 10 illustrates customers having a single DSL line.

Figure 11 illustrates the fields of an exemplary PADI frame.

10 Figure 12 illustrates fields of an Ethernet (and PPPoE) frame.

15 Figure 13 illustrates a payload for the point-to-point protocol over Ethernet (PPPoE) protocol.

Figure 14 illustrates the format of TAGs which may be provided in the payload of a PPPoE protocol frame.

20 Figure 15 illustrates a system in which a customer has a local area network and multiple access (e.g., digital subscriber line) lines.

25 Figure 16 is a bubble diagram which illustrates processes that may be performed in a system which incorporates at least some aspects of the present invention.

30 Figure 17 illustrates processes that may be performed as a part of a line selection process.

Figure 18 illustrates a system in which a first embodiment of the present invention is incorporated.

5 Figure 19 illustrates a machine which may be used to effect various aspects of the present invention.

Figure 20 is a flow diagram of an exemplary line selection method.

10 Figure 21 is an exemplary table data structure which may be used to associate terminals (and sessions) with lines in accordance with the present invention.

15 Figure 22 is a flow diagram of an exemplary connection identification method which may be used with the method of Figure 20 in the context of a system employing the PPPoE protocol.

20 Figure 23 illustrates a system in which a second embodiment of the present invention is incorporated.

25 Figure 24 illustrates an arrangement of Figures 24A and 24B which, collectively, define a communications flow diagram which illustrates an exemplary operation of at least some aspects of the present invention.

S 4. DETAILED DESCRIPTION

30 The present invention involves novel methods, apparatus and data structures for selecting a line for a connection to a service provider. The following description is presented to enable one skilled in the art

to make and use the invention, and is provided in the context of particular applications and their requirements. Various modifications to the disclosed embodiments will be apparent to those skilled in the art, and the general
5 principles set forth below may be applied to other embodiments and applications. Thus, the present invention is not intended to be limited to the embodiments shown and the inventor regards her invention as the following disclosed methods, apparatus and data structures and any
10 other patentable subject matter.

In the following, an environment in which the present invention may operate is described in § 4.1. Then, functions which may be performed by the present invention
15 are described in § 4.2. Then, exemplary processes, architectures, methods and data structures which may be used to effect those functions are described in § 4.3. Examples of operation are then provided in § 4.4. Finally, some conclusions regarding the present invention are set
20 forth in § 4.5.

§ 4.1 ENVIRONMENT IN WHICH THE PRESENT INVENTION MAY OPERATE

25 The present invention may operate in environment such as the one described in § 1.2.4 above with reference to Figure 15. The present invention may operate in the context of a customer premises having an Ethernet based LAN, and in the context of terminals having PPPoE host
30 capabilities (also referred to as a "SHIM") and a service provider having PPPoE access concentrator capabilities.

S 4.2 FUNCTIONS WHICH MAY BE PERFORMED BY THE
PRESENT INVENTION

The present invention may function to minimize or
5 eliminate loop and race conditions at a customer premises
having a number of lines to a communications provider
facility. The present invention may do so by establishing
and/or enforcing line selection policies. The present
invention may also function to allow a customer to
10 provision the use of their lines as they see fit. In this
case as well, the present invention may do so by
establishing and/or enforcing line selection policies.

15 S 4.3 EXEMPLARY PROCESSES, ARCHITECTURE, METHODS
AND DATA STRUCTURES

In the following, processes which may be
performed by the present invention are introduced in
§ 4.3.1 below. Then, architecture, methods and data
20 structures which may be used to effect these processes are
described in the context of two (2) embodiments in §§ 4.3.2
and 4.3.3 below.

25 S 4.3.1 PROCESSES

Figure 16 is a high level bubble diagram of
processes that may be performed by the present invention,
and in particular, which may be performed by an environment
in which the present invention may operate. A number of
30 terminals 1512 may be connected with a hubbing/bridge
process (or more generally, a networking process) 1610,
which may be used to perform hubbing and/or bridging (or
more generally, networking) functions, as described above

for example. The hub/bridge process 1610 may be effected by known hub and/or bridge products.

The local area hub/bridge process 1610 may
5 communicate with a line selection process 1620. The line selection process 1620 may permit a terminal 1512, which can communicate with the hub/bridge process 1610, to establish a session (e.g., a point-to-point protocol session) over a selected one of a number of lines. The
10 line selection process 1620 may select the line to be used based on a policy or policies 1628. This policy(ies) may be put in place (and altered) by a customer using a policy management process 1624. The line selection policy(ies) may be a function of traffic (e.g., number of packets or
15 frames per unit time, and/or number of sessions), past selection states (e.g., to implement a round robin scheme), the terminal 1512 requesting the session, a user requesting the session, a time, a present state of the line(s), and/or a present state of established sessions, etc.

20

A number of lines are terminated by line termination processes 1630 and 1640. The line termination processes 1630 may be effected by known ATU-R units and the line termination processes 1640 may be effected by known
25 ATU-C units. A bridge process 1650 may communicate with the line termination processes 1640 to forward frames towards a connection/session establishment process 1660.

If the terminal and the entity running the
30 connection/session establishment process are to communicate using the point-to-point protocol over Ethernet (PPPoE)

protocol, then the terminal will support a PPPoE host process (SHIM).

Figure 17 is a diagram of processes that may be included in the line selection process 1620. More specifically, physical and data link layer protocol process 1710 may enable communications with the hub/bridge process 1610. A connection state identification process 1720 may be used to determine the state of a connection, such as a PPPoE session connection. Thus, for example, the connection state identification process 1720 may determine whether or not a packet or frame corresponds to the discovery stage (i.e., no session defined yet) or the session stage (i.e., session assigned). If a session has not yet been established (or requested), and therefore, a line has not yet been selected, the frame or packet is forwarded to the policy enforcement process 1730 which may select one of the lines based on a policy or policies 1628.

Given a packet or frame without a defined session (e.g., a discovery stage frame) that has been assigned a line, or a packet or frame having a session, a line forwarding process 1740 is used to forward the packet or frame to the appropriate line, as terminated by an appropriate one of the line termination processes 1630.

Notice that frames or packets destined for the terminal are simply forwarded, from the appropriate one of the line termination processes 1630, directly to the physical and data link layer protocol processes 1710.

Having described processes that may be effected by the present invention, two (2) embodiments of the present invention are now described in §§ 4.3.2 and 4.3.3 below.

5

§ 4.3.2 FIRST EMBODIMENT

An architecture of a first embodiment of the invention is described in § 4.3.2.1 and methods and data structures which may be used in the first embodiment are described in § 4.3.2.2.

§ 4.3.2.1 ARCHITECTURE

15 Figure 18 is a high level block diagram of an exemplary system 1800 which employs at least some aspects of the present invention. As can be appreciated by a comparison of Figures 15 and 18, the present invention provides a line (e.g., a DSL line) selection unit 1810.
20 The line selection unit 1810 may communicate with an (e.g., 100 Base T) Ethernet hub/bridge 1514 and may be used to select a line from the lines 1560 terminated by the ATU-Rs 1516. The line selection unit may have connections to the DSL network via RJ11 jacks. Each RJ-11 jack may support a
25 separate instantiation of a DSL modem. Referring to Figure 19, the line selection unit 1810' may include a processor(s) 1910 (such as a microprocessor, an application specific integrated circuit (or "ASIC"), or a programmable logic array (or "PLA") for example), a storage facility(ies) 1920 (such as RAM, ROM, magnetic storage means, optical storage means, etc. for example),
30 input/output interfaces 1930 (such as a video driver, a

video display, a keyboard, a pointer, a serial port, a parallel port, etc.), and a (e.g., an internal) bus or network 1940. The bus or network 2240 permits the components 1910, 1920 and 1930 to communicate with one another. The line selection unit 1810 may effect the line selection process 1620 (and the policy management process 1624) and may store the policy or policies 1628. The processes may be effected by the processor(s) 1910 executing instructions. The instructions may depend on the policies applied. These policies may be inputted via an input/output interface 1930. The storage facility may store the state of the existing sessions. Having described, at a high level, an architecture which may be used in the first embodiment of the present invention, exemplary methods and data structures are now described in § 4.3.2.2 below.

S 4.3.2.2 METHODS AND DATA STRUCTURES

Figure 20 is a high level flow diagram of an exemplary method 1620' that may be used to effect the line selection process 1620. As shown in block 2010, the LAN line is terminated. This block corresponds to the physical and data link layer protocol processes 1710 of Figure 17. Then, in block 2020, the connection state (of a received packet or frame) is identified. This act corresponds to the connection state identification process 1720 of Figure 17. An exemplary method for performing this act 1720 is described in the context of the PPPoE protocol with reference to Figure 22 later. In this exemplary method 1620', the connection states may be (i) to request the establishment of a new session or packets or frames

associated with the establishment of a new session, (ii) packets or frames associated with an existing session, but not terminating the session, and (iii) packets or frames requesting the termination of an existing session or
5 associated with the termination of an existing session.
Each of these connection states is now addressed, still referring to Figure 20.

As shown in decision block 2030, it is determined
10 whether the connection state is a request for a new session (or packets or frames associated with the establishment of a new session). If the connection state is a request for a new session, a policy is, or policies (recall 1628) are, applied as shown in block 2032, a line is assigned based on
15 the policy or policies as shown in block 2034, and the packet or frame is forwarded to the assigned line as shown in block 2036. The method 1620' is then left via RETURN node 2060.

20 Referring back to block 2034, a line is assigned to a terminal based on the policy or policies. Figure 21 illustrates an exemplary table data structure 2100 for associating a line with a terminal. A column 2110 of the table 2100 includes identifications 2112 of the lines 1560, and a column 2120 of the table 2100 includes an indication 2122 of whether the line is free or whether a terminal 1512 is associated with the line. The identifications 2112 of the lines 1560 may be port numbers of the line selection unit 1810. The indication 2122 of a terminal 1512
25 associated with the line may be a source address (e.g., a MAC address) of the frame or packet. An optional column
30

2130 may include identifications 2132 of sessions supported by the lines 2112.

Referring back to Figure 20, if the connection
5 state is not a request for a new session (or a packet or frame associated with the establishment of a new session), the method 1620' proceeds to decision block 2040 where it is determined whether or not the connection state is an existing session. If not, the method 1620' may report an
10 error, as shown in optional block 2042 and is left via RETURN node 2060. This is done because the connection is either associated with an existing session or it isn't. If, on the other hand, the connection state is an existing session, the method 1620' proceeds to decision block 2050.

15 At decision block 2050, it is determined whether or not the packet or frame is a request to terminate a session or associated with terminating a session. If not, the packet or frame is forwarded to the line assigned to
20 the existing session as shown in block 2056, and the method 1620' is left via RETURN node 2060. (Recall Figure 21.) If, on the other hand, the packet or frame is a request to terminate a session or is associated with terminating a session, the connection is terminated as shown in block
25 2052 and the line assigned to the terminal 1512 and/or session is freed as shown in block 2054 (Recall column 2120 and/or 2130 of table 2100 of Figure 21.), and the method 1620' is left via RETURN node 2060. Thus, any Ethernet frame received from the customer premises 1510 LAN to
30 support a given PPPoE session may be supported over a single embedded modem 1516 for the life of the session.

RECALL FROM BLOCK 2020 OF FIGURE 20 THAT THE CONNECTION STATE IS IDENTIFIED. FIGURE 22 IS A HIGH LEVEL FLOW DIAGRAM OF AN EXEMPLARY METHOD 2020' FOR IDENTIFYING A SESSION STATE IN THE CONTEXT OF THE PPPoE PROTOCOL. AS SHOWN IN BLOCK 2210, CONTROL OR USER DATA IS TAKEN FROM THE FRAME (OR PACKET). (RECALL, ETHER TYPE FIELD 1250' OF THE PPPoE FRAME 1200' AND THE CODE AND SESSION ID FIELDS 1340 AND 1350, RESPECTIVELY, WITHIN THE DATA FIELD 1260.) THE METHOD 2020' THEN PROCEEDS TO DECISION BLOCK 2220. AT DECISION BLOCK 2220, IT IS DETERMINED WHETHER OR NOT THE FRAME (OR PACKET) IS A REQUEST TO ESTABLISH A SESSION. IN THE CONTEXT OF THE PPPoE PROTOCOL, IF THE ETHER TYPE FIELD 1250' IS 0X8863 AND THE CODE FIELD 1340 IS 0X09 (INDICATING THAT THE PACKET IS A PPPoE ACTIVE DISCOVERY INITIATION (PADI) PACKET) (RECALL FIGURE 11.), THE CONNECTION STATE IS SET TO "REQUEST NEW SESSION" AS SHOWN IN BLOCK 2222. THE METHOD 2020' IS THEN LEFT VIA RETURN NODE 2250. RECOGNIZE THAT FURTHER PACKETS ASSOCIATED WITH THE DISCOVERY STAGE (E.G., PADO, PADR AND PADS PACKETS) SHOULD BE ASSIGNED TO THE SAME LINE 1520.. IN THIS WAY, ONCE A SESSION IS REQUESTED, A LINE IS ASSOCIATED, EVEN THOUGH THE SESSION MAY NOT NECESSARILY YET BE ESTABLISHED AND EVEN THOUGH A SESSION ID (RECALL FIELD 1350.) IS NOT NECESSARILY YET ASSIGNED.

25

REFERRING BACK TO DECISION BLOCK 2220, IF IT IS DETERMINED THAT THE FRAME (OR PACKET) IS NOT A REQUEST TO ESTABLISH A SESSION, THE METHOD 2020' PROCEEDS TO DECISION BLOCK 2230. AT DECISION BLOCK 2230, IT IS DETERMINED WHETHER OR NOT THE FRAME (OR PACKET) MERELY CARRIES USER DATA FOR AN ESTABLISHED SESSION (OR WHETHER OR NOT THE FRAME OR PACKET IS A PADO, PADR OR PADS PACKET ASSOCIATED

0
9
8
7
6
5
4
3
2
1
0

with establishing a session in response to a request). If so, the connection state is set to existing session as shown in block 2230 and the method 2020' is left via RETURN node 2250. More specifically, in the context of the PPPoE protocol, if the Ether type field 1250 is set to 0X8864, this indicates that the frame or packet is associated with an existing session (which should already be associated with a line). Furthermore, even if the Ether type field 1250 is set to 0X8863, but the packet is a PADO, PADR or 5 PADS packet, since a line should have already been assigned in response to the receipt of a PADI packet, it is assumed, for the purposes of selecting a line, that a session has 10 already been established although, strictly speaking, the PPPoE protocol is still in the discovery stage.

15 Referring back to decision block 2230, if it is determined that the frame (or packet) does not carry user data for an established session (and the frame or packet is not a PADO, PADR or PADS packet associated with 20 establishing a session in response to a request), the method 2020' proceeds to decision block 2040 where it is determined whether or not the packet or frame is a request to tear down or terminate a session or connection. More 25 specifically, in the context of PPPoE, if the code field 1340 is set to 0XA7, this indicates a request to terminate the session. If it is determined that the packet or frame is a request to tear down or terminate a session or connection, the connection state is set to end session as shown in block 2240, and the method 2020' is left via 30 RETURN node 2250. If, on the other hand, it is determined that the packet or frame is not a request to tear down or terminate a session or connection, the connection state

could not be determined and an error may be reported as shown in optional block 2244.

Having described exemplary architecture, methods 5 and data structures for implementing a first embodiment of the present invention, a second embodiment of the present invention is now described in § 4.3.3 below.

4.3.3 SECOND EMBODIMENT

10

Figure 23 is a high level block diagram of a second embodiment 2300 of the present invention. As can be appreciated by comparing Figures 18 and 23, the main difference between the first and second embodiments is that 15 in the second embodiment 2300, the ATU-Rs 1516' are incorporated into the (DSL) line selection unit 2310. Thus, referring back to Figure 17, the line termination process 1630 would be a part of the line selection process 1620. Otherwise, the operations of the first and second 20 embodiments 1800 and 2300, respectively, would be the same.

The embedded modem is more efficient than utilizing multiple external modems when supporting multiple DSL lines in a premises. This follows because the same DSL 25 line selector logic can keep state for all modems as to what embedded modem a PPPoE session has been assigned. Similarly the same ATM logic (e.g., segmentation and reassembly (or "SAR")) can be used across modems. With separate DSL modems, separate SAR logic may be required. 30 Furthermore, the current state of external ATU-Rs supporting PPPoE may require that a learning bridge

algorithm be supported by the ATU-R. This is not required within the embedded modems.

S 4.4 EXAMPLE OF OPERATION

5

Figure 24 illustrates the arrangement of Figures 24A and 24B which, collectively, define a messaging diagram which illustrates operations of the present invention when used in the context of the PPPoE protocol or a similar 10 protocol. Since illustration of the LAN/bridge process 1610 at the customer premises 1510 and the line termination processes 1640 and bridging process 1650 at the communications provider facility 1520 are not necessary for 15 a good understanding of the present invention, they are not shown in Figure 24 in order to simplify this Figure.

As indicated by communication 2402, the terminal issues a session request which is forwarded (via the LAN/bridge process 1610, not shown) to the connection state identification process 1720. The connection state identification process 1720 identifies the communication as a request for a session and forwards it (and the determined state) to the policy enforcement process 1730 as shown by communication 2406. The policy enforcement process 1730 20 uses information (e.g., state, terminal ID, terminal address, user ID, past state, line state, etc.) to assign one of the lines 1520 based on a policy or policies 1628 (not shown) as indicated by block 2408. The assigned line and its associated terminal ID (e.g., the MAC address of 25 the terminal 1512) are then used to update the table 2100 as shown in communication 2412. The policy enforcement process 1730 may also forward the session request and 30

assigned line to the line forwarding processes 1740 in communication 2414. The line forwarding process 1740 uses the assigned line information in communication 2414 to forward the session request to a line termination process 1630 associated with the assigned line, as shown in communication 2416. The line communication process then forwards the session request (via a line, a line termination process 1640 and a bridge process 1650, not shown) to a session server 1540. Actually, since the session request is broadcast, it can be received by more than one session server 1540 (indicated by ellipses). Indeed, the other terminals 1512 coupled with the LAN hub/bridge 1514 may receive the request.

As indicated by communication(s) 2420, the session server(s) 1540 replies with a session offer(s) which is (are) forwarded (via the bridge process 1650, the line termination process 1640, and the line, not shown) to the line termination process 1630. The line termination process 1630 then forwards the session offer(s) (via the LAN/bridge process 1610, not shown) to the terminal 1512 as shown by communication(s) 2422.

The terminal 1512 then selects which session offer that it will accept. Once a session offer is accepted by the terminal 1512, the terminal 1512 sends a session acceptance communication 2426 with a destination address corresponding to the session server 1540 selected. More specifically, the session acceptance is forwarded (via the LAN/bridge process 1610, not shown) to the connection state identification process. The connection state identification process 1720 then forwards the session

acceptance and its determined connection state to the policy enforcement process 1730. Using the source (e.g., MAC) address or other identification of the terminal 1512, the policy enforcement process looks up the assigned line 5 associated with the terminal 1512 in table 2100, as denoted by 2432. The assigned line is provided to the policy enforcement process 1730 in communication 2434. The policy enforcement process 1730 then forwards the session acceptance and the assigned line to the line forwarding 10 process 1740, as indicated by communication 2436. The line forwarding process 1740 then forwards the session acceptance to a line termination process 1630 associated with the assigned line, as indicated by communication 2438. The line termination process 1630 then forwards the session 15 acceptance (via the line, the line termination process 1640 and the bridge process 1650, not shown) to the session server 1540 whose offer was accepted, as indicated by communication 2440.

20 As indicated by communication 2442, the session server 1540 replies with a session confirmation, which includes a session ID. This reply 2442 is forwarded (via the bridge process 1650, the line termination process 1640, and the line, not shown) to the line termination process 25 1630. The line termination process 1630 then forwards the session acknowledgement (with the session ID) (via the LAN/bridge process 1610, not shown) to the terminal 1512 as shown by communication 2444.

30 At this point, at least in accordance with the PPPoE protocol, the session stage is entered from the discovery stage. During the course of the session, the

terminal 1512 may forward data (with session ID) to the session server 1540 and vice-versa. More specifically, referring to Figure 14B, the terminal 1512 may forward data frames or packets (including a session ID) (via a
5 LAN/Bridge process 1610, not shown) to the connection state identification process 1720, as shown by communication 2448. The connection state identification process 1720 forwards the data frame or packet (with the session ID) and a connection state to the policy enforcement process 1730
10 as indicated by communication 2452. Again, the policy enforcement process 1730 may use the source terminal identification (e.g., the terminal's 1512 MAC address) and/or a session ID to query the table 2100 for an assigned line as indicated by communication 2454. From the table
15 2100, line assigned to the terminal or session is provided as indicated by communication 2456. The policy enforcement process 1730 may then forward the data frame or packet (with the session ID) and the assigned line to the line forwarding process 1740, as indicated by communication
20 2458. The line forwarding process 1740 then forwards the data frame or packet (with the session ID) to a line termination process 1630 associated with the assigned line, as indicated by communication 2460. The line termination process 1630 then forwards the data frame or packet (with
25 the session ID) (via the line termination process 1640 and the bridge process 1650, not shown) to the session server 1540, as indicated by communication 2462.

Further frames or packets of data may be
30 communicated from the terminal 1512 to the session server 1540 in a similar manner.

Regarding communications from the session server 1540 to the terminal 1512, the session server 1540 forwards (via the bridge process 1650, the line termination process 1640, and the line, not shown) a data frame or packet (with session ID) to the line termination process 1630, as indicated by communication 2464. The line termination process 1630 then forwards the data frame or packet (with the session ID) (via the LAN/bridge process 1610, not shown) to the terminal 1512 as shown by communication 2466.

10

Further frames or packets of data may be communicated from the session server 1540 to the terminal 1512 in a similar manner.

15 With the PPPoE protocol, either the terminal 1512 or the session server 1540 may terminate a session. If the terminal 1512 initiates the termination of the session, it may forward a session termination request (with the session ID) (via the LAN/bridge process 1610, not shown) to the connection state identification process 1720, as indicated by communication 2470b. If the session server 1540 initiates the termination of the session, it may first be forwarded to the terminal 1512, in a manner similar to the other communications from the session server 1540 to the terminal 1512, described above, as shown by communication 2470a. The terminal 1512 may then forward the session termination request (with the session ID) (via the LAN/bridge process 1610, not shown) to the connection state identification process 1720. In any event, the connection state identification process 1720 will forward, at least the determined state and terminal identification (e.g., terminal 1512 MAC address) and/or session ID, to the policy

enforcement process 1730. Once both the terminal 1512 and the session server 1540 have finished any necessary communications related to terminating the session (and connection), the policy enforcement process 1730 may send a 5 request to update the table 2100 as indicated by communication 2472. The table 2100 may be updated by freeing the line assigned to the session.

In the forgoing description and examples, notice
10 that establishing a session and terminating a session may often involve multiple communications between the terminal 1512 and session server 1540. In a preferred embodiment of the present invention, the policy enforcement process 1730 assigns a line and updates the table 2100 to reflect this
15 assignment upon receipt of the first communication requesting to establish a session. Subsequent communications related to establishing a session (e.g., offer, acceptance and acknowledgement messages) may be treated as normal communications of an established session.
20 Similarly, in a preferred embodiment of the present invention, the policy enforcement process 1730 frees a line and updates the table 2100 to reflect this fact upon receipt of the last communication requesting to terminate a session or connection. Prior communications related to
25 terminating a session may be treated as normal communications of an established session.

§ 4.5 CONCLUSIONS

30 The line selection devices of the present invention may be user configurable to allow session distribution policies to be configured in it via a

management interface. (This interface could be web based or via an RS-232 serial console interface.) There are numerous policies that can be defined. The following list provides some examples:

5

Round robin --each PPPoE session that arrives from the premises LAN is assigned a (DSL) line in round robin order.

Least loaded -- An arriving PPPoE session is assigned to
10 the (DSL) line that has the least of amount of traffic load over the past time interval, where the time interval may be user definable (e.g., 5 minutes).

60
59
58
57
56
55
54
53
52
51
50
49
48
47
46
45
44
43
42
41
40
39
38
37
36
35
34
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1

Static assignment -- An arriving PPPoE session from a given
15 terminal (e.g., PC) is always assigned to the same (DSL) line. The terminal's MAC address may be used as the identifier to assess to which line the session is assigned.

Benefits of the line selection devices of the
20 present invention include the following. First, it allows businesses to take statistical advantage of their (DSL) lines. This follows since one (1) line per terminal (e.g., PC) is not required. Second, it allows the customer to define the policy to determine how the (DSL) lines are
25 utilized. Third, it allows existing LAN equipment to be used with multiple (DSL) line implementations. More specifically, it eliminates adverse effects of using multiple (DSL) lines with an Ethernet-based infrastructure. Fourth, it allows (DSL) line redundancy to be easily
30 supported as the algorithms can be structured to ignore lines that have failed when assigning new PPPoE sessions. Finally, it achieves economies of scale by allowing the

same ATM and control logic to support multiple embedded modems.

03625000 02500